Growing semantically meaningful models for visual SLAM

Alex Flint*, Christopher Mei†, Ian Reid*, and David Murray*
*Active Vision Lab
†Mobile Robotics Group
Dept. Engineering Science
University of Oxford, Parks Road, Oxford, UK
{alexf,cmei,ian,dwm}@robots.ox.ac.uk

Abstract

Though modern Visual Simultaneous Localisation and Mapping (vSLAM) systems are capable of localising robustly and efficiently even in the case of a monocular camera, the maps produced are typically sparse point–clouds that are difficult to interpret and of little use for higher–level reasoning tasks such as scene understanding or human–machine interaction. In this paper we begin to address this deficiency, presenting progress on expanding the competency of visual SLAM systems to build richer maps. Specifically, we concentrate on modelling indoor scenes using semantically meaningful surfaces and accompanying labels, such as "floor", "wall", and "ceiling" — an important step towards a representation that can support higher-level reasoning and planning.

We leverage the Manhattan world assumption and show how to extract vanishing directions jointly across a video stream. We then propose a guided line detector that utilises known vanishing points to extract extremely subtle axis–aligned edges. We utilise recent advances in single view structure recovery to building geometric scene models and demonstrate our system operating on–line.

1. Introduction

The simultaneous localisation and mapping problem has received considerable attention over past decades, which is unsurprising given its centrality in fields from mobile robotics to augmented reality. Considerable progress has been made over this period and modern SLAM systems perform efficiently and robustly even in the case of a monocular video stream [7].

Many high–level reasoning and planning problems can benefit from an accurate underlying SLAM system, but SLAM point clouds alone provide a poor basis upon which to reason about scene semantics as they represent just a fraction of the information present in the original images. Photometric cues for edges, surfaces, occlusion boundaries, and texture information, among others, are lost entirely.

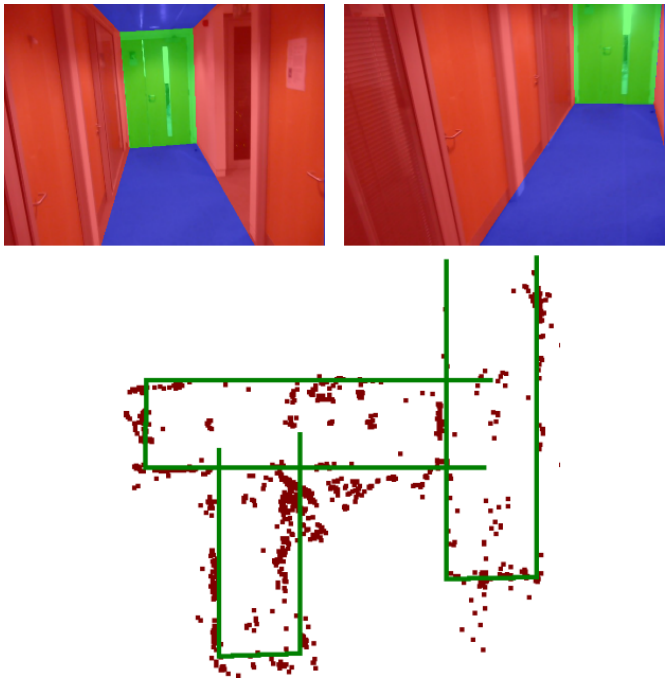

Figure 1. The floor plan and two camera views of the Manhattan building model reconstructed automatically and on–line for the "lab" sequence. The point cloud is so sparse that even a human would struggle to reconstruct this floor plan using point data alone. Our system uses rich photometric cues together with camera poses recovered from SLAM to reason about Manhattan world geometry.

Consider the top–down view of a SLAM point cloud depicted in Figure 1. The points are sparse and non–uniformly distributed; even a human would have difficulty identifying the location of walls using the point cloud alone. The SLAM system used here has located map points at salient

corners in the images, which are optimal for camera localisation but unhelpful for inferring high–level scene structure.

An important step towards higher–level reasoning tasks is to represent the structure of the scene at a semantic level, using meaningful concepts such as "floor" and "wall". These entities assist in reasoning as they correlate with, for example, the locations in which objects might appear or the set of feasible actions in a given location.

The present work investigates the extraction of such a semantic scene model from video sequences, utilising an underlying SLAM system together with rich photometric cues. We focus on indoor scenes as they exhibit a rich set of regularities that assist in model building. One such regularity is the prevalence of three mutually orthogonal orientations around which man–made environments are often built up. The use of this observation to restrict the space of possible scene interpretations has come to be known in the literature as the Manhattan world assumption. Typically, the orientation of the camera with respect to these dominant directions is *a priori* unknown and must be extracted explicitly, for example by identifying vanishing points. We propose a new method that leverages the camera poses provided by SLAM to estimate vanishing directions jointly across a video sequence, allowing frames with salient edge information to inform the system about vanishing directions in frames with poor or non–existent edge information.

We then return to the images to re–identify line segments using the known vanishing points to inform our search. We propose a novel line detector that takes vanishing point locations into account to identify important structural edges that are aligned with a dominant direction but which may exhibit weak gradients, whilst ignoring stronger gradients generated by surface texture or occlusion boundaries.

The final component of our system joins the line segments into Manhattan building structures, inspired by recent work in single view reconstruction [9]. We enumerate possible building structures that the observed line segments could generate under the Manhattan world assumption and evaluate each for consistency with surface orientations estimated from photometric cues. The key contribution of this section is the extension to multiple views of the hypothesis testing framework, which allows our system to disambiguate scenarios that the single view approach would fail on.

The remainder of this paper is organised as follows. In the next section we overview prior work in this field. Following this we describe the three primary components of our system in order: the joint vanishing point estimator, the Manhattan line search algorithm, and the reconstruction system, with results given in each section. Finally we discuss the results and present closing remarks.

2. Background

In recent years the need for semantics to be connected with SLAM maps has been recognised by several researchers. Stachniss *et al*. [15] have taken an image–centric approach wherein 3D features are projected into frames and used together with photometric cues to classify environments into semantic categories such as "corridor" or "room". Posner *et al*. [11] take this a step further by segmenting incoming frames into semantic categories based jointly on 3D and photometric cues. Xiao and Quan [18] have approached this problem by solving a multiple label MRF over superpixels from two or more views. Brostow *et al*. [1] demonstrate that several intuitive 3D features are sufficient for semantic video segmentation.

Buschka and Saffiotti [2] opt instead to reason directly in the map. They build an occupancy grid and identify room boundaries use morphological filters. More recently, Golovinskiy *et al*. [4] learn to segment and identify objects in city–wide reconstructions using machine learning techniques. These approaches discard the original images after building a map, which we believe throws away many useful cues not captured in the map.

Furukawa *et al*. [3] have shown how to reconstruct Manhattan environments using graph cuts. Their reconstructions are of high quality but their quoted computation times (of between one minute and one hour forty minutes) make their approach unsuitable to on–line computation. Their system uses multiple–view stereo to first reconstruct a dense point cloud, whereas we are interested in working with a sparse point cloud and leveraging photometric cues for on–line performance.

Several authors have recently demonstrated impressive single view reconstruction systems. Hoiem *et al*. [6] pose the problem as a multi–class segmentation problem, with labels corresponding to 3D geometry, while Saxena *et al*. [13] obtain reconstructions by estimating surface normals of image patchlets. Gould *et al*. [5] reason simultaneously about geometry and object labels.

Lee *et al*. [9] take a geometric approach in which detected line segments are connected to form 3D building hypotheses. The authors show that an exhaustive search over building hypotheses is feasible since Manhattan world models are highly constrained. This work forms the basis for Section 5, which extends this approach to multiple views.

Many researchers have proposed methods for completing partial lines or identifying subtle, yet semantically important, line segments that humans see easily. For example, Sarti *et al*. [12] iteratively fill missing boundaries starting from a reference point and Shufelt [14] models line detection errors explicitly. This body of literature deals with the single image scenario, whereas the emphasis of our work is on leveraging metric SLAM information to improve the accuracy and speed of line detection.

3. Extracting a canonical coordinate frame

Input to our system is a set of key–frames sampled from the video sequence, with SLAM poses P_i for each frame. The sub–sampling of key–frames from the video stream is determined by the SLAM implementation [7]; for a typical exploration sequence, key–frames are added at intervals of around 1–3 seconds. P_i consists of a rotation R_i and translation $\boldsymbol{t}_i$. These are measured with respect to some coordinate frame determined during initialisation, which we will refer to as the "SLAM coordinate frame".

In order to make use of the Manhattan world assumption we must first discover the orientation of the Manhattan world with respect to the camera. Equivalent to the Manhattan world assumption is the statement that there exists a "canonical" coordinate frame in which world surfaces are axis–aligned. The problem is therefore to discover the rotation R_w between the canonical coordinate frame and the SLAM coordinate frame. R_w is constant for all frames since the SLAM system has already determined the relative transformations between successive frames.

In the past researchers have discovered R_w for single images by identifying vanishing points [8, 14]. This approach fails for frames that do not contain edges in at least two of the Manhattan directions, or that contain predominantly non–axis–aligned edges. Both scenarios are common in video sequences of indoor environments since the camera often views only a small portion of the scene. Since R_w is fixed for all frames it makes sense to leverage all available data during estimation rather than to estimate vanishing points separately for each frame.

In the structure–from–motion setting it has been proposed to recover R_w by clustering surface normals estimated from local neighbourhoods in the point cloud [3]. The surface normal approach requires a dense scene reconstruction, whereas the point cloud provided by on–line SLAM is too sparse to obtain dense orientation estimates. Though we could have implemented dense reconstruction on top of SLAM, we show in the remainder of this section that R_w can be estimated very robustly from line detections and camera poses alone — an approach that is also far less computationally intensive.

We begin by running the Canny edge detector on each key–frame, followed by an edge linking algorithm [8] to identify a set of straight line segments $L_j = \{\boldsymbol{x} : \boldsymbol{l}_j^T \boldsymbol{x} = 0\}$.

The projections of the three vanishing points into the i^{th} frame are related to R_w by

$$\boldsymbol{v}_1 = R_i R_w \boldsymbol{e}_1 \quad (1)$$

$$\boldsymbol{v}_2 = R_i R_w \boldsymbol{e}_2 \quad (2)$$

$$\boldsymbol{v}_3 = R_i R_w \boldsymbol{e}_3 \quad (3)$$

where $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$ are unit vectors in the x, y, and z directions respectively. We can now write down an error function to be minimised in terms of R_w:

$$f(R_w) = \sum_{i,j,k} r_{jk} \left(\frac{{\boldsymbol{l}_j}^T R_i R_w \boldsymbol{e}_k}{\sqrt{{l_{jx}}^2 + {l_{jy}}^2}} \right)^2 , \quad (4)$$

where r_{jk} is the responsibility of the k^{th} vanishing point for the j^{th} line segment. The squared term in (4) is the algebraic deviation[1] of the j^{th} line segment from the k^{th} vanishing point in frame i, and the full error (4) is the sum over all such deviations, weighted by the respective responsibilities.

While other authors search for vanishing points by clustering on the Gaussian sphere, enforcing orthogonality constraints afterwards, we prefer to optimise in terms of R_w directly, which embeds the orthogonality constraint into the estimation process itself.

We now describe an EM algorithm to optimise R_w with respect to (4). During the E step we compute the responsibilities r_{jk} of each vanishing point $\boldsymbol{v}_k$ for each line segment $\boldsymbol{l}_j$. We assume a Gaussian likelihood

$$p(\boldsymbol{l}_j \mid \boldsymbol{v}_k) = G\left(\frac{{\boldsymbol{l}_j}^T \boldsymbol{v}_k}{\sqrt{{l_{jx}}^2 + {l_{jy}}^2}}; \sigma \right) \quad (5)$$

as well as a fixed prior on observing a spurious line segment

$$p(S_j) = \rho \, . \quad (6)$$

Noting that we must have

$$p(S_j) + \sum_{i=1}^{3} r_{ji} = 1 \quad (7)$$

and assuming that line segments are equally likely to be associated with any of the three vanishing points, we have

$$r_{jk} = \frac{p(\boldsymbol{l}_j \mid \boldsymbol{v}_k)}{\alpha + \sum_i p(\boldsymbol{l}_j \mid \boldsymbol{v}_i)} \quad (8)$$

where we have substituted

$$\alpha = \frac{3\rho}{1-\rho} p(\boldsymbol{l}_j \mid S_j) \, . \quad (9)$$

The M step consists of optimising R_w with respect to the error function (4). There is no closed form solution for the optimal R_w so we instead perform gradient descent. We represent R_w in the Lie algebra as a member of the special orthogonal group $SO(3)$, so

$$R_w = \exp(\sum m_i G_i) \quad (10)$$

[1] We use the algebraic deviation rather than the re–projection error [10] for simplicity and because we have found that the very large number of line segments we obtain from the entire video sequence renders a more complicated error metric unnecessary.

where the G_i are the generator matrices for $SO(3)$ and the m_i provide a minimal representation for the 3D rotation matrix group. The advantage of using this representation is that at each step we are guaranteed that R_w remains a pure rotation, whereas under other representations, such as optimising the elements of the 3×3 rotation matrix directly, this is not the case. Differentiating (4) with respect to $\boldsymbol{m}$ yields

$$\nabla f = \sum_{i,j,k} \frac{2{r_{jk}}^2 \boldsymbol{l}_j{}^T R_i R_w e_k \, \boldsymbol{l}_j{}^T R_i \nabla R_w \boldsymbol{e}_k}{{l_{jx}}^2 + {l_{jy}}^2} \quad (11)$$

$$\nabla R_w = [G_1 \boldsymbol{e_1}, G_2 \boldsymbol{e_2}, G_3 \boldsymbol{e_3}] \, . \quad (12)$$

Our update rule is then

$$\boldsymbol{m^{t+1}} = \boldsymbol{m^t} - \frac{f(\boldsymbol{m^t})}{\|\nabla f(\boldsymbol{m^t})\|_2} \nabla f(\boldsymbol{m^t}) \, . \quad (13)$$

In summary, to obtain R_w we iterate between assigning responsibilities (the E step) and optimising R_w given those responsibilities (the M step). Each M step consists of a gradient descent in the Lie algebra. In practice we found that our system converged in around 25 iterations of the EM algorithm, and that approximately 10 steps were required for each gradient descent. Since the gradient descent algorithm converged quickly and robustly we found no need to use higher–order approaches such as the Gauss–Newton algorithm.

Figure 2 shows the vanishing points identified in one of our sequences. Since each frame is informed by the entire sequence we are able to identify a globally consistent coordinate frame where single–image vanishing point detection fails. Figure 3 shows a side–by–side comparison with the single–image vanishing point detector of [8]. Recently proposed improvements to the single–image approach [16] may improve slightly on these, but we found that in cases where the single–image approach fails there is often simply not enough information available in individual frames to identify the appropriate coordinate frame, so any single–image approach will necessarily fail.

3.1. Identifying the vertical direction

Of the three dominant directions defined by R_w, two correspond to horizontal directions and the third to the vertical direction. The latter is semantically distinct since it defines the orientation of the ground and ceiling planes, as well as the direction in which gravity operates. It is easy to identify the vertical axis since humans necessarily move over the ground plane when capturing video sequences, and have limited scope for moving the camera in the up–down direction. We therefore set the vertical axis to that over which camera positions range the least. Having identified R_w there are only three possible choices, and we found this heuristic to work correctly in all of our evaluation sequences.

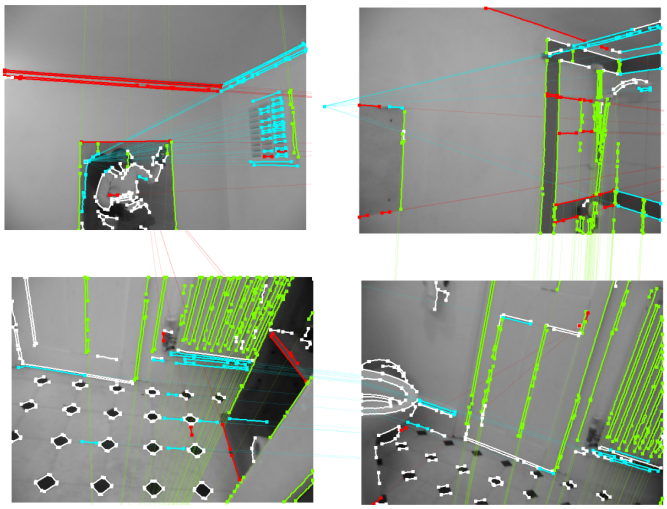

Figure 2. Four frames from the "bathroom" sequence and the detected vanishing points. The vanishing points are correctly identified despite the strong distractor gradients generated by the floor tiles, which is possible only by integrating information from multiple views into the estimation process. Vertical lines are green, horizontal lines are blue (along the room's longer dimension) and red (along the room's shorter axis).

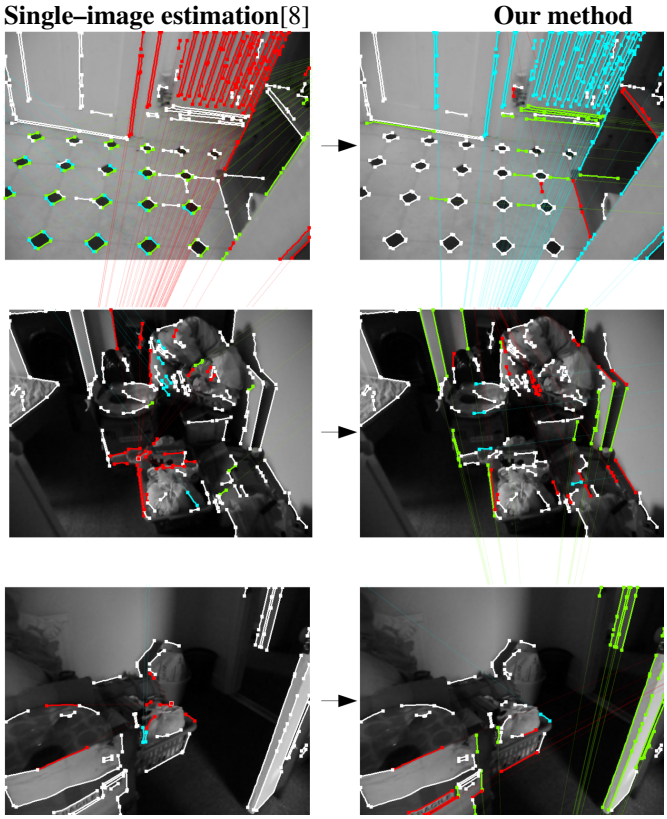

Figure 3. Comparison between vanishing points estimated for single views (left column) and joint estimates from 20 frames in a video sequence (right column). Our method is able to identify vanishing directions correctly in these difficult cases, whereas the single view estimator is confused by non–Manhattan line segments.

3.2. Relaxing the Manhattan world assumption

The strong Manhattan assumption states that any pair of surfaces of interest are either parallel or orthogonal to one another. One common deviation from this is scenes with walls that are orthogonal to the ground and ceiling but not to one another. We define the weak Manhattan assumption as "the environment consists of a horizontal ground plane and corresponding ceiling plane, and a set of vertical wall segments extending continuously between them." Weakly Manhattan environments contain much of the regularity of strongly Manhattan environments. We deal with the weak Manhattan assumption as follows. First, we run the EM algorithm described above to obtain R_w. Next, for each line $\boldsymbol{l}_j$ marked as spurious by the EM algorithm we find its intersection with the horizon,

$$\boldsymbol{u}_j = R_w^{-T} R_i^{-T} \boldsymbol{l}_j \times \boldsymbol{e}_3 \ , \tag{14}$$

which would be its vanishing point if it were horizontal in the world. Vertical surfaces of a given orientation will generate identical $\boldsymbol{u}_j$ (modulo measurement error), so we may identify additional vertical orientations by clustering the intersections $\{\boldsymbol{u}_j\}$. We adopt a voting algorithm in which we parametrise $\boldsymbol{u}_j$ in terms of the angle θ_j about the z axis

$$\theta_j = \text{atan}(\boldsymbol{e}_2{}^T \boldsymbol{u}_j,\ \boldsymbol{e}_1{}^T \boldsymbol{u}_j) \ . \tag{15}$$

We accumulate the θ_j into histogram bins and identify any local maxima θ_i^* above a threshold k. Each θ_i^* represents a cluster of line segments corresponding to an additional vertical orientation. Finally, we re–estimate the vanishing point for each cluster by minimising the likelihood (5) via least–squares.

4. Manhattan line search

Many of the structurally important edges in indoor scenes generate weak intensity gradients in comparison to those generated by surface texture or occlusion boundaries. For example, if two adjoining walls are painted the same colour then the intensity gradient at their intersections will be generated only from subtle Lambertian lighting effects. Hence the most important edges (for reconstruction purposes) are often the most difficult to identify. Three examples of this phenomenon are shown in Figure 6.

We have found that the Canny edge detector, which is the near–universal approach to edge detection within the vision literature, misses many structurally important edges unless the thresholds are lowered to such a point that any textured surface generates many thousands of spurious line segments. Conversely, when the Canny thresholds are set to large enough values in order that spurious detections are kept to a manageable level then the true positive line segments are insufficient for the the reconstruction process described in the following section (although they *are* sufficiently numerous to identify the canonical coordinate frame since the space of rotations has fewer degrees of freedom than the space of building structures).

To overcome this we return to the images after determining the canonical coordinate frame and perform a second search for Manhattan line segments.[2] with the known vanishing points informing the search. For each pixel $\boldsymbol{x}$ we begin by estimating its vanishing point association. The orthogonality constraint guarantees that no two vanishing points will be very close to one another, so we obtain a reliable estimate using only the local image gradient $\boldsymbol{g}$:

$$\text{assoc}(\boldsymbol{x}) = \text{argmin}_i \frac{(\boldsymbol{x} - \boldsymbol{v}_i)^T \boldsymbol{g}}{\|\boldsymbol{x} - \boldsymbol{v}_i\|} \ , \tag{16}$$

where i ranges over the three possible vanishing points.

Each vanishing point is associated with a one–parameter family of lines extending from it (three cases are shown in Figure 4) and Manhattan line segments correspond to clusters of pixels lying along a single vanishing line. We propose the following voting strategy to identify Manhattan line segments. We parametrise each pixel $\boldsymbol{x}$ by the angle $\theta(i, \boldsymbol{x})$ that its vanishing line $\boldsymbol{m} = \boldsymbol{x} \times \boldsymbol{v}_i$ makes about the associated vanishing point $\boldsymbol{v}_i$,

$$\theta(i, \boldsymbol{x}) = \text{atan}(\boldsymbol{x}^T \boldsymbol{v}_j, \boldsymbol{x}^T \boldsymbol{v}_k) \tag{17}$$

where all vectors are in homogeneous coordinates. This representation is free of singularities and sampling uniformly in θ space produces a uniform distribution of lines in the image.

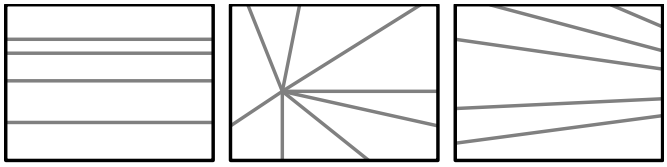

Figure 4. Lines meeting at a vanishing point.

Next we build a histogram over θ for each of the three vanishing points. Each vote is weighted by the strength of the local image gradient and the agreement between the gradient orientation and the direction to the associated vanishing point. Formally, the weight with which a pixel $\boldsymbol{x}$ votes for the vanishing line $\boldsymbol{m} = \boldsymbol{x} \times \boldsymbol{v}_i$ is

$$w_{\boldsymbol{x},i} = \|\boldsymbol{g}\|^\alpha \ \left(\frac{(\boldsymbol{x} + \boldsymbol{g})^T \boldsymbol{m}}{\|\boldsymbol{x} + \boldsymbol{g}\|_2 \sqrt{{m_x}^2 + {m_y}^2}} \right)^\beta \ , \tag{18}$$

where $\boldsymbol{g}$ is the image gradient at $\boldsymbol{x}$ and α and β are parameters that determine the relative importance of the gradient orientation and magnitude.

[2] Manhattan line segments are those generated by surfaces oriented in one of the dominant directions.

The bin width for the histograms is set to the minimum size such that no bin spans more than two pixels anywhere in the image. Histogram peaks are identified by applying non–maximum suppression followed by thresholding. The final line segments are identified by walking along the vanishing lines corresponding to peaks in the histogram and linking edge pixels using hysteresis. A line segment is started each time a gradient magnitude above the high threshold k_0 is detected, and is ended when the gradient magnitude drops below the low threshold k_1. This algorithm requires just one pass over the image to populate all three histograms. Identifying peaks in the histogram and walking along the corresponding lines is then computationally trivial.

Compared to the cascaded Hough transform of Tuytelaars *et al*. [17], our approach estimates vanishing points from multiple views simultaneously and leverages orthogonality constraints for robustness. Our line search is linear in the size of the image, whereas each iteration of the cascaded Hough transform has complexity cubic in the image size.

Figure 5 shows four example frames and the lines corresponding to the histogram peaks. The detector fires at subtle axis–aligned gradients while ignoring strong but non–axis–aligned distractors. Figure 6 shows a side–by–side comparison with the line detector of [8], which employs the standard Canny edge detector followed by an edge linking algorithm. In each example our detector is able to identify important structural edges that the Canny detector does not respond to. We found that lowering the Canny thresholds sufficiently to detect these edges generated many thousands of spurious line segments on the textured carpet and other areas.

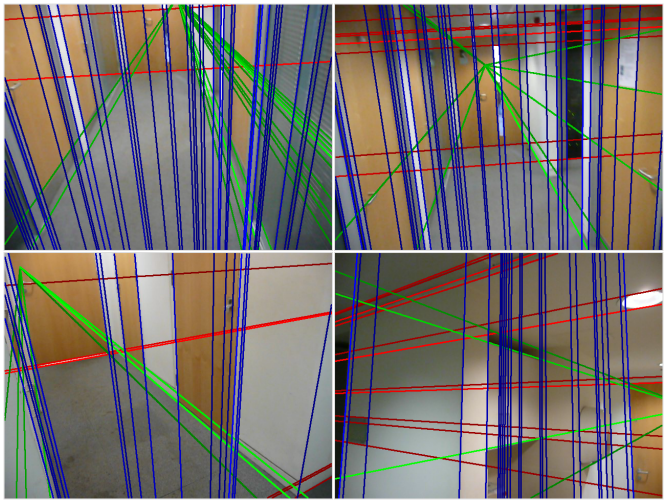

Figure 5. Four frames from the "lab" sequence with peaks in the the histograms over θ highlighted, each of which corresponds to a Manhattan line in the image. The rays capture the important geometric structure extremely accurately, with almost no false positives.

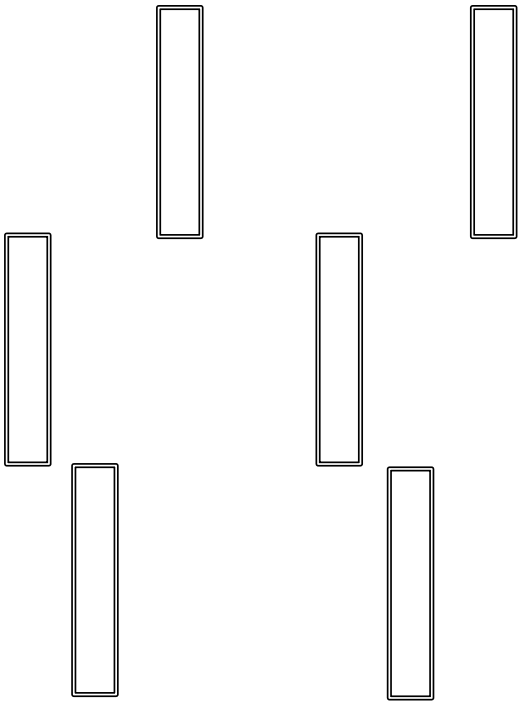

Figure 6. Comparison between the Canny/edge linking detector (left column) and our guided line search (right column). Our approach is able to recover several subtle yet structurally important line segments that Canny misses.

5. Recovering building structures

In the previous sections we used the pose estimates provided by the SLAM system to find stable structural lines in an axis–aligned frame. The final component of our system shows how this information can be combined to build a semantically meaningful reconstruction of the environment. In [9], the authors proposed a branch-and-bound algorithm that exploits the Manhattan assumption to build a reconstruction for a single monocular image. In this section we extend this approach to evaluate building hypotheses using multiple views in order to leverage the metric camera trajectories provided by the SLAM system.

The assumption of a Manhattan world containing infinite floor and ceiling planes constrains possible scene interpretations considerably. The entire set of feasible building structures can be efficiently enumerated using the following branch–and–bound algorithm. First, each valid pair of horizontal lines initialises a building hypothesis with a single wall and no corners. Next, corners are added recursively by intersecting detected line segments with existing building hypotheses, up to a maximum depth D. Not all hypotheses are physically realisable but those that are not can be easily identified using simple heuristics [9].

Each building hypothesis $\mathcal{B}$ defines a unique 3D model up to an unknown scale factor s^*. To determine s^* we lever-

age the observation that some SLAM landmarks will fall on the surfaces we are trying to reconstruct. We therefore propose the following voting scheme. For each SLAM landmark visible in the current frame we identify the surface it falls upon within $\mathcal{B}$. Next we compute the scaling $\hat{s}$ such that the reconstructed surface contains that 3D point. Each SLAM landmark then votes for the scale it induces on $\mathcal{B}$. We accumulate votes into a histogram and set s^* to the scale that received the greatest number of votes.[3]

Knowledge of s^* permits a full 3D reconstruction and hence allows transfer of building structure between frames. We test each building hypothesis according to its consistency with surface orientation estimates in the current frame and the K preceding frames. The orientation estimates are computed separately for each frame using the line sweep approach of [9]. The score for a building hypothesis $\mathcal{B}$ is computed as the total number of pixels in all $K+1$ frames for which the orientation predicted by $\mathcal{B}$ agrees with the orientation estimate. The hypothesis with greatest score is output as the final model.

6. Results

We tested our approach on several videos of indoor environments, estimating a building structure for each key–frame. To allow the SLAM system to continue real–time operation the estimation was performed in a parallel thread. Figure 8 shows processing times per key–frame. Processing time varies from 0.045 to 3.4 seconds as a result of the varying number of line detections, but note that processing time does not increase systematically as the map expands.

Figure 7 shows the model reconstructed using our approach for the "lab" sequence. The second row shows the initial orientation estimates obtained using the line sweep algorithm. These estimates are noisy, inconsistent, and in several instances major surfaces are missed entirely. However, by leveraging metric SLAM information and reasoning in multiple views simultaneously our approach is able to generate the correct model in all frames, as shown in the third row of Figure 7. Notice that this is despite the fact that only one frame views both the floor and ceiling simultaneously — which is a prerequisite for the single–image approach of Lee *et al*.

[3] An alternate approach would be to directly hypothesise building structures in 3D, avoiding the need to recover s^*. However, this would entail first reconstructing each observed line segment in 3D, which is unattractive because of the need to identify correspondences between lines in consecutive frames and would render our system sensitive to a single mislocalisation. Our approach replaces the difficult line reconstruction problem with the more constrained problem of identifying s^*. Lee *et al*. ignore s^* since they build reconstructions for single images only.

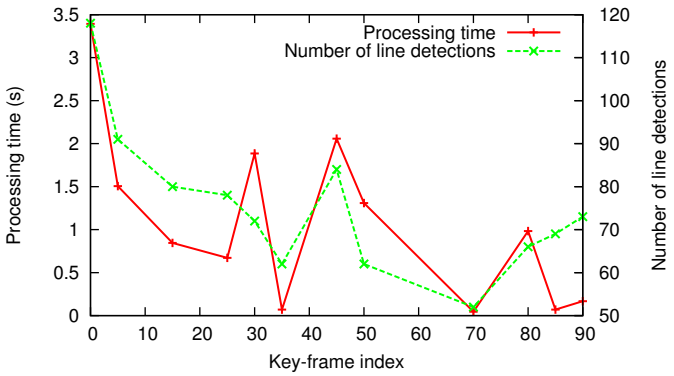

Figure 8. Processing time and number of line detections for each key frame. Processing time varies with the number of line detections but does not increase as the map gets larger.

7. Conclusion

We have shown that semantically meaningful models of indoor scenes can be generated on–line. By combining pose information generated by an underlying SLAM system together with photometric cues not present in the point cloud, and incorporating the Manhattan world assumption, we can reason across multiple views accurately and efficiently to infer scene structure even when the SLAM point cloud is very sparse. This work represents an important step towards using SLAM in higher-level reasoning tasks for which point clouds alone are unsuited. In particular, knowledge about scene geometry can be beneficial for object detection, and by locating geometric primitives in the canonical coordinate frame we could automatically learn about common configurations such as doors and windows. More generally, we hope to explore the idea of contextual priming for on–line applications such as augmented reality and robotic navigation.

Acknowledgements This work has partly been supported by the European Commission under grant agreement number FP7-231888-EUROPA. We gratefully acknowledge the support of the EPSRC through grant EP/D0370771/1.

References

[1] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 44–57, Berlin, Heidelberg, 2008. Springer-Verlag.

[2] P. Buschka and A. Saffiotti. A virtual sensor for room detection. In *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, volume 1, pages 637–642 vol.1, 2002.

[3] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Manhattan-world stereo. *Computer Vision and Pat-*

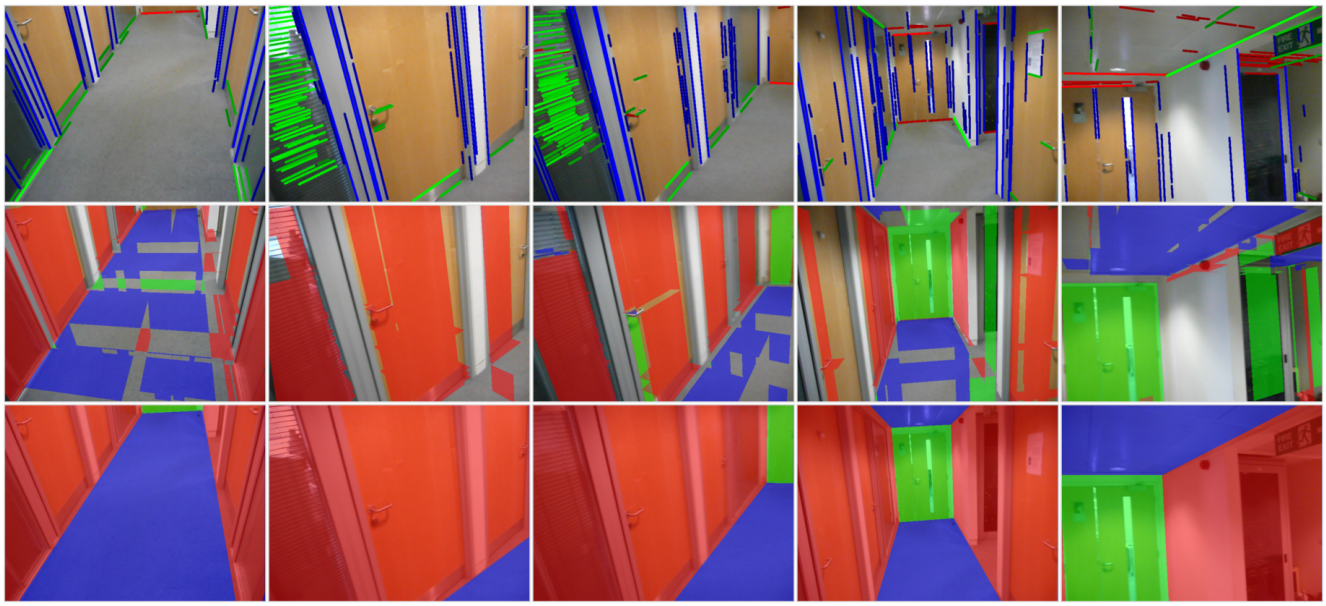

Figure 7. The final reconstruction for the "lab" sequence. The top row shows the line segments identified by the guided line search; the middle row shows the surface orientation estimates from the individual frames; the bottom row shows the final model projected into five frames. The orientation estimates shown in the second row are noisy and incomplete but we are able to obtain an accurate model by combining information from multiple views. The reconstruction accurately captures the primary surfaces within the environment.

tern Recognition, IEEE Computer Society Conference on, 0:1422–1429, 2009.

[4] A. Golovinskiy, V. G. Kim, and T. Funkhouser. Shape-based recognition of 3d point clouds in urban environments. In *Proc 12th IEEE Int Conf on Computer Vision*, volume 2, 2009.

[5] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Proc 12th IEEE Int Conf on Computer Vision*, volume 2, 2009.

[6] D. Hoiem, A. A. Efros, and M. Hébert. Geometric context from a single image. In *Proc 10th IEEE Int Conf on Computer Vision*, pages 654–661, 2005.

[7] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[8] J. Kosecká and W. Zhang. Video compass. In *Proc 7th European Conf on Computer Vision*, volume 2353 of *Lecture Notes in Computer Science*, pages 4: 476–490. Springer, 2002.

[9] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009.

[10] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 482–488, Jun 1998.

[11] I. Posner, D. Schroeter, and P. Newman. Online generation of scene descriptions in urban environments. *Robot. Auton. Syst.*, 56(11):901–914, 2008.

[12] A. Sarti, R. Malladi, and J. A. Sethian. Subjective surfaces: A geometric model for boundary completion. *International Journal of Computer Vision*, 46(3):201–221, 2002.

[13] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009.

[14] J. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(3):282–288, Mar 1999.

[15] C. Stachniss, O. Martnez-Mozos, A. Rottmann, and W. Burgard. Semantic labeling of places. In *in Proceedings of the International Symposium on Robotics Research*, 2005.

[16] J.-P. Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *Proc 12th IEEE Int Conf on Computer Vision*, volume 2, 2009.

[17] T. Tuytelaars, M. Proesmans, and L. J. V. Gool. The cascaded Hough transform as support for grouping and finding vanishing points and lines. In *AFPAC '97: Proceedings of the International Workshop on Algebraic Frames for the Perception-Action Cycle*, pages 278–289, London, UK, 1997. Springer-Verlag.

[18] J. Xiao and L. Quan. Multiple view semantic segmentation for street-view images. In *Proc 12th IEEE Int Conf on Computer Vision*, volume 2, 2009.